Amendments to the Claims

1. (previously amended) A system for simulating machine instructions on a host machine comprising:

a monitor executing in a direct execution environment on the host machine that translates the machine instructions into translated code, the monitor modifying original values of segment information in a descriptor table to force intervention by the monitor when the translated code references a memory access, thereby preventing ~~prevent~~ the translated code from being accessed, and thereby preventing the translated code from being modified;

a virtual machine executing in the direct execution environment on the host machine that executes the translated code stored in memory in a simulated operating system; and

a kernel executing in the direct execution environment on the host machine that detects exceptions occurring in the virtual machine and transfers control between the virtual machine and the monitor according to a type of the exceptions,

wherein an operating system executing in a host environment on the host machine also supports a full platform simulator that includes device models for a target processor, wherein the machine instructions are specific to the target processor, the monitor to execute translated code comprising sensitive instructions in an auxiliary simulator, and when the translated code are to access a simulated device, the full platform simulator executing the translated code that represents the device access ~~simulated operating system code to be executed on the virtual machine~~, and wherein results of executing the translated code are provided to a user.

2. (canceled)

3. (previously amended) The system of claim 1 wherein the translated code and the original machine instructions access the memory using a same set of addresses.

4. (canceled)

5. (original) The system of claim 1 wherein the monitor replaces one of the machine instructions with a capsule if the machine instruction accesses a system state of a central processing unit of the host machine.

6. (canceled)

7. (previously amended) The system of claim 1 wherein the monitor modifies the descriptor table to remove a portion of a segment that overlaps with the memory storing the translated code.

8. (previously amended) The system of claim 1 wherein the monitor modifies the descriptor table to replace a segment with a substitute segment, which, when accessed, causes an exception to be generated.

9. (currently amended) A method of simulating machine instructions on a host machine comprising:

translating the machine instructions into translated code, by a monitor executing in a direct execution environment on the host machine;

storing the translated code in memory;

executing the translated code in a virtual machine in the direct execution environment;

preventing the translated code from being modified, by modifying original values of segment information in a descriptor table which force intervention by the monitor when the translated code references a memory access;

detecting exceptions in the execution of the translated code;

transferring control to an appropriate simulation module on the host machine according to a type of the exceptions, wherein when the translated code comprises a sensitive instruction, executing the translated code in an auxiliary simulator in the monitor, and when the translated code is to access a simulated device, executing the translated code in a full platform simulator executing in a host environment on the host machine, and otherwise, executing the translated

code in a simulated operating system in the virtual machine in the direct execution environment; and

providing results of executing the translated code to a user,

wherein an operating system executing in the host environment on the host machine ~~also~~ supports ~~a~~ the full platform simulator for a target processor, wherein the machine instructions are specific to the target processor, ~~that~~ wherein the full platform simulator includes simulation modules and device models~~, the simulator executing the translated code that represents simulated operating system code to be executed on a virtual machine~~.

10. (original) The method of claim 9 further comprising simulating a device.

11. (previously amended) The method of claim 9 further comprising accessing memory by the translated code using a same set of addresses as a set of addresses used by the original machine instructions.

12. (original) The method of claim 9 further comprising replacing one of the machine instructions with a capsule if the machine instruction accesses a system state of a central processing unit of the host machine.

13. (previously amended) The method of claim 9 wherein the modifying of the descriptor table is to prevent the translated code from being modified, the descriptor table including attributes of a segment of the memory.

14. (previously amended) The method of claim 13 further comprising modifying the descriptor table to remove a portion of a segment that overlaps with the memory storing the translated code.

15. (original) The method of claim 13 further comprising modifying the descriptor table to replace the segment with a substitute segment, which, when accessed, causes an exception to be generated.

16. (currently amended) A computer program product for simulating machine instructions, the program product residing on a machine readable medium comprising instructions for causing a host machine to:

translate a set of machine instructions into translated code, by a monitor executing in a direct execution environment on the host machine;

store the translated code in memory;

execute the translated code in a virtual machine in the direct execution environment;

prevent the translated code from being modified, by modifying original values of segment information in a descriptor table which forces intervention by the monitor when the translated code references a memory access;

detect exceptions in the execution of the translated code;

transfer control to an appropriate simulation module on the host machine according to a type of the exceptions, wherein when the translated code comprises a sensitive instruction, execute the translated code in an auxiliary simulator in the monitor, and when the translated code is to access a simulated device, execute the translated code in a full platform simulator executing in a host environment on the host machine, and otherwise, execute the translated code in a simulated operating system in the virtual machine in the direct execution environment; and

provide results of executing the translated code to a user,

wherein an operating system executing in the host environment on the host machine ~~also~~ supports ~~a~~ the full platform simulator for a target processor, wherein the machine instructions are specific to the target processor ~~that~~ wherein the full platform simulator includes device models, ~~the simulator executing the translated code that represents simulated operating system code to be executed on a virtual machine~~.

17. (previously amended) The computer program product of claim 16 further comprising instructions for causing the host machine to simulate a device.

18. (previously amended) The computer program product of claim 16 further comprising instructions for causing the host machine to access memory by the translated code using a same set of addresses as a set of addresses used by the original machine instructions.

19. (previously amended) The computer program product of claim 16 further comprising instructions for causing the host machine to replace one of the machine instructions with a capsule if the machine instruction accesses a system state of a central processing unit of the host machine.

20. (previously amended) The computer program product of claim 16 wherein instructions for causing the host machine to modify a descriptor table is to prevent the translated code from being modified, the descriptor table including attributes of a segment of the memory.

21. (previously amended) The computer program product of claim 20 further comprising instructions for causing the host machine to modify the descriptor table to remove a portion of a segment that overlaps with the memory storing the translated code.

22. (previously amended) The computer program product of claim 20 further comprising instructions for causing the host machine to modify the descriptor table to replace a segment with a substitute segment, which, when accessed, causes an exception to be generated.

23. (currently amended) A system for simulating an instruction set architecture on a platform comprising:

a virtual machine monitor on the platform to translate machine instructions of a target processor, the target processor having a first instruction set architecture, into translated code of a second instruction set architecture, the target processor to run in a virtual machine on the platform, the virtual machine monitor to modify original values of segment information in a descriptor table to prevent the translated code from being accessed by forcing intervention by the virtual machine monitor when the translated instructions reference a memory access, thereby preventing the translated code from being modified;

the virtual machine to execute the translated code stored in memory; and

a virtual machine kernel on the platform to detect exceptions occurring in the virtual machine and to transfer control between the virtual machine and the virtual machine monitor according to a type of the exceptions,

wherein ~~an~~ a host operating system executing ~~on~~ in a host ~~machine~~ environment on the platform supports a full platform simulator that includes device models for the target processor, wherein the machine instructions are specific to the first instruction set architecture of the target processor, the virtual machine monitor executing translated code comprising sensitive instruction in an auxiliary simulator, and when the translated code are to access a simulated device, the full platform simulator executing the translated code that represents the device access, ~~the simulator executing the translated code that represents simulated operating system code of the first instruction set architecture to be executed on the virtual machine in the second instruction set architecture, thereby allowing the target processor to be simulated without disturbing the operating system running on the host machine~~, and wherein results of the execution of the translated code are provided to a user.

24. (previously presented) The system of claim 23, wherein the translated code and the original machine instructions access the memory using a same set of addresses.

25. (canceled)

26. (previously presented) The system of claim 23 wherein the virtual machine monitor is to replace one of the machine instructions with a capsule if the machine instruction accesses a system state of a central processing unit of the host machine, the capsule being one of a simple capsule and a complex capsule, and wherein simple capsule is executed by the virtual machine and a complex capsule is executed by the virtual machine monitor.

27. (previously presented) The system of claim 23, wherein the virtual machine monitor modifies the descriptor table to remove a portion of a segment that overlaps with the memory storing the translated code.

28. (previously presented) The system of claim 23, wherein the virtual machine monitor modifies the descriptor table to replace a segment with a substitute segment, which, when accessed, causes an exception to be generated.